LiMoE: Mixture of LiDAR Representation Learners from Automotive Scenes

Supplementary Material

Table of Contents

A. Additional Implementation Details

In this section, we provide additional details to facilitate the implementation and reproducibility of the methods within the proposed LiMoE framework.

A.1. Datasets

In this work, we conduct extensive experiments across a diverse set of LiDAR semantic segmentation datasets to validate the effectiveness of the proposed LiMoE framework.

- **nuScenes** [5, 9] is a large-scale, multimodal dataset designed for autonomous driving, featuring six cameras, five radars, one LiDAR, along with IMU and GPS sensors. The dataset comprises $1,000$ driving scenes collected in Boston and Singapore. For the point cloud semantic segmentation task, it provides 1.4 billion annotated points across $40,000$ point clouds, with each LiDAR point labeled into one of 32 semantic categories. The point clouds are captured using a Velodyne HDL-32E LiDAR sensor. In this work, a mini-train split is created from the full training set for model pretraining during the Image-to-LiDAR and CML stages, adhering to the SLidR protocol [26]. For the SMS stage, the training set is further split to generate subsets containing 1%, 5%, 10%, 25%, and 100% of annotated scans for fine-tuning. More details about this dataset can be found at https://nuscenes.org/nuscenes.
- **SemanticKITTI** [1] is a large-scale benchmark dataset tailored for semantic scene understanding in autonomous driving. The dataset was collected using a Velodyne HDL-64E LiDAR sensor, capturing diverse real-world scenarios such as urban traffic in city centers, residential neighborhoods, highways, and rural countryside roads around Karlsruhe, Germany. The dataset consists of 22 densely labeled point cloud sequences derived from the KITTI Odometry benchmark [10], with each point annotated into one of 28 semantic categories. In this work, the training set is uniformly split to create a subset with 1% of the scans for fine-tuning. More details about this dataset can be found at https://semantic-kitti.org.
- **Waymo Open** [29] is a large-scale, high-quality, and diverse dataset designed to advance perception in autonomous driving. The dataset features multimodal data collected using five high-resolution cameras and five LiDAR sensors. It includes $1,150$ driving scenes recorded across a variety of suburban and urban areas, captured at different times of the day to ensure diversity in lighting, weather, and traffic conditions. For the LiDAR semantic segmentation task, each point in the dataset is annotated into one of 23 semantic categories. In this work, the training set is uniformly split to create a subset with 1% of the scans for fine-tuning. More details about this dataset can be found at https://waymo.com/open.
- **ScribbleKITTI** [31] is a weakly supervised variant of the SemanticKITTI [1] dataset, designed to advance research in semantic scene understanding with minimal annotation effort. Unlike SemanticKITTI, which provides dense, point-wise annotations for every LiDAR point, ScribbleKITTI employs sparse line scribble annotations as a cost-effective alternative. This approach drastically reduces annotation requirements, with the dataset containing approximately 189 million labeled points – around 8.06% of the fully supervised dataset – resulting in a 90% reduction in annotation time. In this work, the training set is uniformly split to create subsets with 1% and 10% of the scans for fine-tuning. More details about this dataset can be found at https://github.com/ouenal/scribblekitti.
- **RELLIS-3D** [11] is a multimodal dataset curated for semantic scene understanding in complex off-road environments. It consists of five traversal sequences collected

along three unpaved trails on the RELLIS Campus of Texas A&M University. For the LiDAR semantic segmentation task, point-wise annotation was generated by projecting image-based semantic labels onto the point cloud using precise camera-LiDAR calibration. Each LiDAR point is categorized into one of 20 semantic categories. In this work, the training set is uniformly split to create subsets with 1% and 10% of the scans for fine-tuning. More details about this dataset can be found at http://www.unmannedlab.org/research/RELLIS-3D.

- **SemanticPOSS** [22] is a small-scale benchmark dataset designed for semantic segmentation, with a particular focus on dynamic instances in real-world off-road environments. The dataset was captured using a Hesai Pandora LiDAR sensor, a forward-facing color camera, and four wide-angle mono cameras. The data was collected on the campus of Peking University. SemanticKITTI comprises 7 sequences, with each point annotated into one of 14 semantic categories. In this work, we adopt sequences 00 and 01 as half of the annotated training scans and sequences 00 to 05 (excluding 02 for validation) to create the full set of annotated scans for fine-tuning. More details about this dataset can be found at https://www.poss.pku.edu.cn/semanticposs.html.
- **SemanticSTF** [33] is a LiDAR point cloud dataset specifically designed to enable robust perception under adverse weather conditions, which is derived from the STF benchmark [3]. The dataset was collected using a Velodyne HDL-64 S3D LiDAR sensor and includes a diverse set of 2,076 scans captured across various weather conditions: 694 snowy, 637 dense-foggy, 631 light-foggy, and 114 rainy scans. Each point in the dataset is labeled with one of 21 semantic categories. In this work, the training set is uniformly split to create subsets with 50% and 100% of the scans for fine-tuning. More details about this dataset can be found at https://github.com/xiaoaoran/SemanticSTF.
- **SynLiDAR** [32] is a synthetic LiDAR dataset generated from various virtual environments. The dataset was created using the Unreal Engine 4 platform, capturing diverse outdoor scenarios such as urban cities, towns, harbors, *etc*. It consists of 13 LiDAR sequences with a total of 198,396 scans, with each point labeled into one of 32 semantic categories. In this work, the training set is uniformly split to create subsets with 1% and 10% of the scans for fine-tuning. More details about this dataset can be found at https://github.com/xiaoaoran/SynLiDAR.
- **DAPS-3D** [12] consists of two subsets: DAPS-1 and DAPS-2, both captured by a Ouster OS0 LiDAR sensor. DAPS-1 is semi-synthetic, generated to simulate various real-world cleaning tasks, while DAPS-2 was captured during a real field trip of a cleaning robot operating in the VDNH Park in Moscow. In this work, the training set from the DAPS-1 subset is uniformly split to create subsets with 50% and 100% of the scans for fine-tuning. More details about this dataset can be found at https://github.com/subake/DAPS3D.
- **Synth4D** [25] is a synthetic dataset captured using a simulated HDL LiDAR sensor within the CARLA simulator. The dataset consists of two subsets, collected from a vehicle navigating through four distinct scenarios: town, highway, rural area, and city. In this work, the training set from the Synth4D-nuScenes subset is uniformly split to create subsets with 1% and 10% of the scans for fine-tuning. More details about this dataset can be found at https://github.com/saltoricristiano/gipso-sfouda.
- **nuScenes-C** [13] is a dataset within the Robo3D benchmark, specifically designed to evaluate the robustness of 3D detectors and segmentors under out-of-distribution scenarios and natural corruptions commonly encountered in real-world environments. The dataset incorporates eight types of corruptions: "fog", "wet ground", "snow", "motion blur", "beam missing", "crosstalk", "incomplete echo", and "cross-sensor" scenarios. Each corruption type is simulated following physical principles or engineering guidelines and includes three severity levels: light, moderate, and heavy. More details about this dataset can be found at https://github.com/ldkong1205/Robo3D.

A.2. Training Configuration

In this subsection, we present the implementation details of the LiMoE framework, which is organized into three stages.

- **Image-to-LiDAR Pretraining** focuses on transferring knowledge from image representations to LiDAR point clouds. This stage builds on the methodologies of SLidR [26] and SuperFlow [35]. We employ the ViT [8] architecture as the image backbone, pretrained using DINOv2 [21], with three variants: Small, Base, and Large. Input images are resized to 224×448 and augmented with random horizontal flipping. For the LiDAR-based backbone, we select FRNet [36], MinkUNet-34 [7], and SPVCNN [30], corresponding to the range, voxel, and point representations, respectively. Point cloud augmentations include random flipping along horizontal and vertical axes (with a 50% probability), rotation along the z-axis within the range of $-180°$ to $180°$, and scaling with a factor sampled uniformly from $[0.95, 1.05]$. The LiDAR-based networks are pretrained using eight GPUs for 50 epochs with a batch size of 4 per GPU. We initialize the learning rate to 0.01 and employ the AdamW optimizer [19] with a OneCycle scheduler [28].
- **Contrastive Mixture Learning (CML)** promotes the in-

tegration of diverse LiDAR representations into a unified feature space. In this stage, the pretrained range, voxel, and point networks are mixed through a Mixture of Experts (MoE) layer, leveraging their complementary strengths to form a cohesive single-representation network. To enhance representation diversity, LiDAR point clouds are augmented with varied parameters, generating multiple respective views for each representation. The network is pretrained on eight GPUs for 50 epochs, with a batch size of 4 per GPU. The initial learning rate is set to 0.001, and training utilizes the AdamW optimizer [19] with a OneCycle scheduler [28]. The pseudo-code for CML is detailed in Algorithm A.

- **Semantic Mixture Supervision (SMS)** aims to improve downstream segmentation performance by fusing semantic logits from multiple representations under semantic label supervision. For individual representation supervision, the range network is optimized using Cross-Entropy loss, Lovasz-Softmax loss [2], and Boundary loss [24] with weights of 1.0, 2.0, and 1.0, respectively. The voxel network employs Cross-Entropy loss, Lovasz-Softmax loss [2], weighted at 1.0 and 2.0, while the point network relies solely on Cross-Entropy loss. The MoE-fused logits are supervised exclusively with Cross-Entropy loss. The training is conducted on four GPUs for 100 epochs, with a batch size of 4 per GPU. The initial learning rate for each representation's backbone is set to 0.001, and 0.01 for all other parameters. The AdamW optimizer [19] and a OneCycle scheduler [28] are used for optimization. The pseudo-code for SMS is detailed in Algorithm B.

A.3. Evaluation Configuration

To evaluate the semantic segmentation performance across various semantic classes, we employ the widely used Intersection-over-Union (IoU) metric. The IoU score for a specific class is computed as follows:

$$\text{IoU} = \frac{TP}{TP + FP + FN}, \tag{1}$$

where TP (True Positive) denotes the number of points correctly classified as belonging to the class, FP (False Positive) denotes the number of points incorrectly classified as belonging to the class, and FN (False Negative) denotes the number of points belonging to the class but misclassified as another class. To assess overall segmentation performance, we report the mean IoU (mIoU), calculated as the average IoU across all semantic classes.

To evaluate robustness, we adopt the Corruption Error (CE) and Resilience Rate (RR) metrics, following the setup established in Robo3D [13]. The CE and RR for a specific corruption type are computed as follows:

$$\text{CE} = \frac{\sum_{i=1}^{3}(1 - \text{IoU}_i)}{\sum_{i=1}^{3}(1 - \text{IoU}_i^{\text{base}})}, \quad \text{RR} = \frac{\sum_{i=1}^{3}\text{IoU}_i}{3 \times \text{IoU}_{\text{clean}}}, \tag{2}$$

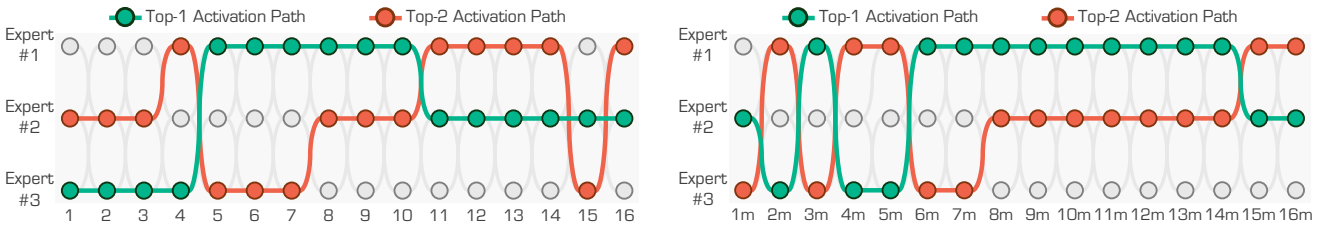
Algorithm A CML, PyTorch-stype

```
# Point2Range: convert point cloud to range image
# Point2Voxel: convert point cloud to sparse voxel
# Range2Point: project range image to point cloud
# Voxel2Point: project sparse voxel to point cloud
# Group: Group features according to superpoint
# P: point cloud with shape (N, L)
# SP: superpoint
# B_R, B_V, B_P: Range-view, sparse voxel, and point
    network
# B_S: Student network for distilling
# D: output channel for each representation network
# Cont: contrastive learning function

class MoE(nn.Module):

   def __init__(self, channels):
      super(MoE, self).__init__()
      self.fusion = nn.Linear(channels*3, channels)

      self.w_gate = nn.Parameter(
         torch.zeros(channels, 3),
         requires_grad=True)
      self.w_noise = nn.Parameter(
         torch.zeros(channels, 3),
         requires_grad=True)

      self.softplus = nn.Softplus()
      self.softmax = nn.Softmax(1)

   def forward(self, range_feats, voxel_feats,
       point_feats):
      # feature alignment
      range_feats = Range2Point(range_feats)
      voxel_feats = Voxel2Point(voxel_feats)
      fusion_feats = torch.cat(
         [range_feats, voxel_feats, point_feats],
         dim=-1)
      fusion_feats = self.fusion(fusion_feats)

      clean_logits = feats @ self.w_gate
      raw_noise_stddev = feats @ self.w_noise
      noise_stddev = self.softplus(raw_noise_stddev)
      noise_logits = torch.randn_like(clean_logits)
          * noise_stddev
      logits = clean_logits + noise_logits
      gates = self.softmax(logits) # (N, 3)
      alpha, beta, gamma = gates[:, 0:1], gates[:,
          1:2], gates[2:3]
      return alpha * range_feats + beta *
          voxel_feats + gamma * point_feats

moe_layer = MoE(D)
R = Point2Range(P) # (H, W, L)
V = Point2Voxel(P) # (M, L)
F_R, F_V, F_P = B_R(R), B_V(V), B_P(P)
moe_feats = moe_layer(F_R, F_V, F_P)
student_feats = B_S(P)
# generate superpoint embedding
K = Group(moe_feats, SP)
Q = Group(student_feats, SP)
# loss function
loss = Cont(K, Q)
```

where $\text{IoU}_i^{\text{base}}$ denotes the IoU score of the baseline model for the corresponding corruption severity, and $\text{IoU}_{\text{clean}}$ indices the IoU score on the "clean" evaluation set. To measure overall robustness, we report the mean CE (mCE) and mean RR (mRR), which are calculated as the average CE and RR values across all corruption types.

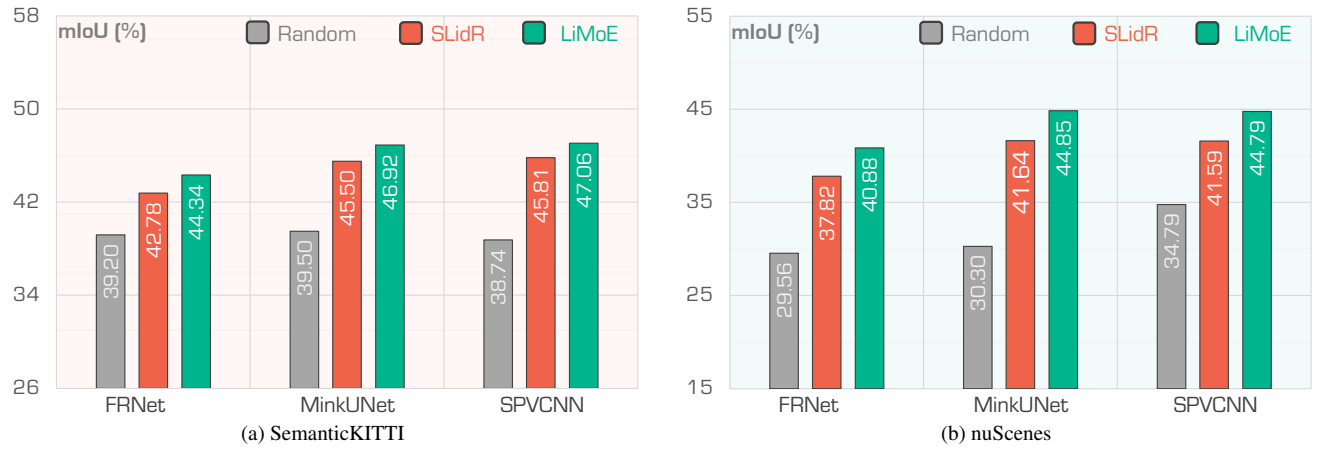
B. Additional Quantitative Results

In this section, we present class-wise LiDAR semantic segmentation results to reinforce the findings and conclusions



(a) Expert Activation Paths (Beam Number) (b) Expert Activation Paths (Distances to Ego)

Figure A. Visual interpretations of the expert activation paths in CML. The experts are #1 range view, #2 voxel, and #3 point, respectively.



(a) SemanticKITTI (b) nuScenes

Figure B. **Ablation study on different backbones** for downstream tasks. The backbones are initialized with random weights, SLidR [26], and LiMoE, respectively, and fine-tuned on the SemanticKITTI [1] and nuScenes [9] datasets using 1% annotations.

presented in the main body of the paper.

B.1. Class-Wise Linear Probing Results

Tab. D showcases class-wise LiDAR semantic segmentation results on the *nuScenes* [5, 9] dataset, achieved through pretraining followed by linear probing. The evaluation covers all 16 semantic classes, offering a detailed performance comparison across diverse object categories. LiMoE consistently surpasses single-representation baselines for every class, including challenging categories like "pedestrian", "bicycle", and "traffic cone". These results emphasize the advantage of our approach in utilizing complementary features from range images, sparse voxels, and raw points during the CML stage to capture high-level semantic correlations effectively.

B.2. Class-Wise Fine-Tuning Results

Tab. E presents class-wise LiDAR semantic segmentation results on the *nuScenes* [5, 9] dataset, derived from pretraining followed by fine-tuning with only 1% of the available annotations. The results highlight that LiMoE consistently outperforms single-representation baselines across all classes, with particularly notable gains for dynamic objects such as "pedestrian", "bicycle", and "motorcycle", which often exhibit complex structures. These improvements stem from the SMS stage, where the integration of multiple representations enables the model to capture complementary object attributes, enhancing segmentation performance.

Table A. **Detection comparison** of state-of-the-art pretraining methods pretrained and fine-tuned on the *nuScenes* dataset [5], using specified data proportions. All methods utilize CenterPoint [38] as 3D object detection backbones. All scores are given in percentage (%). The best scores are highlighted in **bold**.

Method	Venue	nuScenes					
		5%		10%		20%	
		mAP	NDS	mAP	NDS	mAP	NDS
Random	-	38.0	44.3	46.9	55.5	50.2	59.7
PointContrast [34]	ECCV'20	39.8	45.1	47.7	56.0	-	-
GCC-3D [16]	ICCV'21	41.1	46.8	48.4	56.7	-	-
SLidR [26]	CVPR'22	43.3	52.4	47.5	56.8	50.4	59.9
TriCC [23]	CVPR'23	44.6	54.4	48.9	58.1	50.9	60.3
CSC [6]	CVPR'24	45.3	54.2	49.3	58.3	51.9	61.3
SuperFlow [35]	ECCV'24	46.0	54.9	49.7	58.5	52.5	61.5
+ LiMoE	**Ours**	**47.3**	**55.3**	**50.6**	**59.0**	**53.2**	**61.8**

B.3. 3D Object Detection

To further evaluate the effectiveness of LiMoE, we extend our framework to the 3D object detection task. Specifically, we integrate three heterogeneous representation experts and distill their knowledge into VoxelNet [37] dur-

Algorithm B SMS, PyTorch-stype

```
# Point2Range: convert point cloud to range image
# Point2Voxel: convert point cloud to sparse voxel
# Range2Point: project range image to point cloud
# Voxel2Point: project sparse voxel to point cloud
# P: point cloud with shape (N, L)
# Y: point cloud semantic label with shape (N)
# B_R, B_V, B_P: Range-view, sparse voxel, and point
    network
# C: number of classes
# CE: loss function between gt and predict logits

class MoE(nn.Module):

    def __init__(self, channels):
        super(MoE, self).__init__()
        self.fusion = nn.Linear(channels*3, channels)

        self.w_gate = nn.Parameter(
            torch.zeros(channels, 3),
            requires_grad=True)
        self.w_noise = nn.Parameter(
            torch.zeros(channels, 3),
            requires_grad=True)

        self.softplus = nn.Softplus()
        self.softmax = nn.Softmax(1)

    def forward(self, range_feats, voxel_feats,
        point_feats):
        # feature alignment
        range_feats = Range2Point(range_feats)
        voxel_feats = Voxel2Point(voxel_feats)
        fusion_feats = torch.cat(
            [range_feats, voxel_feats, point_feats],
            dim=-1)
        fusion_feats = self.fusion(fusion_feats)

        if self.training:
            clean_logits = feats @ self.w_gate
            raw_noise_stddev = feats @ self.w_noise
            noise_stddev = self.softplus(
                raw_noise_stddev)
            noise_logits = torch.randn_like(
                clean_logits) * noise_stddev
            logits = clean_logits + noise_logits
        else:
            logits = clean_logits
        gates = self.softmax(logits) # (N, 3)
        alpha, beta, gamma = gates[:, 0:1], gates[:,
            1:2], gates[2:3]
        return alpha * range_feats + beta *
            voxel_feats + gamma * point_feats

moe_layer = MoE(C)
R, Y_R = Point2Range(P), Point2Range(Y) # (H, W, L)
V, Y_V = Point2Voxel(P), Point2Voxel(Y) # (M, L)
L_R, L_V, L_P = B_R(R), B_V(V), B_P(P)
moe_logits = moe_layer(F_R, F_V, F_P)
# loss function
loss = CE(L_R, Y_R) + CE(L_V, Y_V) + CE(L_P, Y) + CE
    (moe_logits, Y)
```

ing the CML stage. For downstream fine-tuning, we follow the detection pipeline of CenterPoint [38]. As shown in Tab. A, our method achieves substantial improvements over single-representation learning, further demonstrating the effectiveness of MoE in unifying multiple representations into a compact and expressive feature space.

B.4. Representation Diversity

To investigate the role of representation diversity, we conduct an ablation study by replacing the three heterogeneous experts in our framework with three identical sparse voxel-based models, all implemented using MinkUNet [7]. The results, summarized in Tab. B, show that using multiple identical experts provides only a marginal improvement over single-representation learning. However, this setting remains significantly inferior to LiMoE, which integrates diverse representations. This performance gap underscores the importance of representation diversity: by leveraging complementary features from range images, sparse voxels, and raw points, the MoE layer effectively captures richer geometric and semantic information, leading to superior segmentation performance.

Table B. Ablation study on **incorporating representation diversity**. All scores are given in percentage (%).

Method	nuScenes LP	nuScenes 1%	nuScenes 5%	KITTI 1%	Waymo 1%
Random	8.10	30.30	47.84	39.50	39.41
SLidR [26]	45.35	41.64	55.83	45.50	48.32
3 × MinkUNet	45.51	42.72	56.73	46.35	48.94
LiMoE	**46.56**	**46.89**	**58.09**	**47.96**	**49.50**

Table C. Ablation study on **mixing strategies** for integrating multiple representations. All scores are given in percentage (%).

Mixing	nuScenes LP	nuScenes 1%	nuScenes 5%	KITTI 1%	Waymo 1%
Random	8.10	30.30	47.84	39.50	39.41
SLidR [26]	45.35	41.64	55.83	45.50	48.32
Concatenate	45.82	44.75	56.43	46.76	48.53
Addition	45.73	44.82	56.83	46.40	48.71
Average	46.56	46.89	57.12	47.96	49.04
LiMoE	**46.56**	**46.89**	**58.09**	**47.96**	**49.50**

B.5. Effectiveness of MoE-Based Mixing

In this work, we employ MoE to selectively aggregate complementary features from multiple representations. However, alternative feature mixing strategies, such as concatenation, addition, and averaging, can also be considered. To validate the effectiveness of our MoE-based approach, we conduct an ablation study comparing different mixing strategies, with results summarized in Tab. C. Our method achieves the best performance, as it effectively acts as an attention-based mechanism, dynamically selecting the most relevant features for each LiDAR point. In contrast, the other three mixing strategies treat all representations equally, lacking the ability to adaptively capture complementary features, which limits their effectiveness.

B.6. Extend to Different Backbones

We evaluate the downstream performance of FRNet [36], MinKUNet [7], and SPVCNN [30] across various pre-training strategies, including random initialization, SLidR [26], and LiMoE. Importantly, the downstream fine-tuning

does not employ the MoE strategy to combine multiple LiDAR representations, ensuring a fair comparison among the methods. Fine-tuning is conducted on two widely-used datasets, SemanticKITTI [1] and nuScenes [9], with only 1% of the annotations available. As shown in Fig. B, LiMoE pretraining consistently improves the performance of single-representation methods. This demonstrates the scalability and generalizable feature representations learned during the LiMoE pretraining stage, making it effective for enhancing downstream tasks.

C. Additional Qualitative Results

In this section, we provide additional qualitative examples to visually compare different approaches presented in the main body of the paper.

C.1. Route Activations from CML

LiDAR sensors inherently operate with a fixed number of beams, resulting in a structured arrangement of data points within the captured point clouds. This beam-based configuration provides a natural attribution for the LiDAR point clouds, with each beam contributing a distinct set of points that collectively form a comprehensive 3D representation of the environment. Furthermore, the distance of points from the ego vehicle often correlates with the orientation and elevation of the laser beams. Upper beams are typically designed to detect objects at longer distances, capturing information about the far-field surroundings. In contrast, middle beams are optimized for medium-range detections, while lower beams primarily focus on capturing close-proximity objects [14, 15].

To understand how each LiDAR representation contributes to the fused features within the MoE layer, we conduct a detailed statistical analysis of export selection patterns during the CML stage. Specifically, we examined the activation frequency of each representation – range view, voxel, and point – across varying laser beams and distances from the ego vehicle, measuring their respective contributions to the fused outputs. As depicted in Fig. A, distinct focus areas emerge for each LiDAR representation, aligning with their inherent strengths. The range view representation shows a higher activation frequency in middle-range regions. The voxel representation demonstrates a significant focus on upper laser beams and far-field regions. The point representation dominates in close-range regions. This analysis underscores the complementary nature of the three representations. The MoE layer dynamically selects the most suitable representation based on the spatial and distance characteristics of the input, enabling a more robust and comprehensive understanding of 3D environments.

C.2. Point-Wise Activation from SMS

SMS supervises the feature learning process by integrating the semantic logits from multiple LiDAR representations with guidance from semantic labels. To illustrate which object attributes each representation focuses on within the LiDAR point clouds, we analyze the contribution of each representation to the semantic logits during the SMS stage.

Specifically, the MoE layer computes a gating score that indicates the relative importance of each representation for individual points. We highlight the most relevant attributes contributed by each representation and project them onto the corresponding point clouds. As shown in Fig. C, Fig. D, Fig. E, and Fig. F, the range view representation predominantly emphasizes dynamic objects, such as "car", "truck", as well as objects in medium-range regions. The voxel representation excels in capturing static background elements, such as "road" and far-field objects within sparse regions. The point representation specializes in intricate details, such as object edges and close-range features, which are crucial for accurate boundary delineation.

This visualization demonstrates the complementary nature of these representations and underscores the effectiveness of SMS in dynamically leveraging their unique strengths. By aligning these diverse features, SMS ensures comprehensive feature learning, leading to improved segmentation performance across varied object types and environmental conditions.

C.3. LiDAR Segmentation Results

In Fig. H, Fig. I, Fig. J, and Fig. K, we present qualitative LiDAR segmentation results, highlighting the performance of models pretrained on the *nuScenes* [41] dataset using various methods and fine-tuned on the *SemanticKITTI* dataset with 1% of the available annotations. As depicted, LiMoE consistently outperforms single-representation approaches by capturing intricate scene details and achieving a significant reduction in segmentation errors across challenging semantic classes. Notably, it excels in handling dynamic objects such as "pedestrian", where other methods often struggle. These results highlight the ability of our multi-representation fusion framework to integrate complementary features, leading to more robust and accurate segmentation.

C.4. Cosine Similarity Results

In Fig. G, we provide additional cosine similarity maps generated during the CML stage. These maps demonstrate the ability of LiMoE to align features from different LiDAR representations, showcasing high semantic correlations across diverse regions of the scene during pretraining. This alignment reflects the effectiveness of our framework in fusing information from range images, sparse voxels, and raw points to capture complementary semantic

cues. By fostering strong inter-representation consistency, our method establishes a solid foundation for downstream tasks, improving the performance and reliability of LiDAR-based segmentation systems in real-world scenarios.

D. Public Resources Used

In this section, we acknowledge the use of public resources, during the course of this work.

D.1. Public Codebase Used

We acknowledge the use of the following public codebase, during the course of this work.

- MMEngine[1] ..... Apache License 2.0
- MMCV[2] ..... Apache License 2.0
- MMPretrain[3] ..... Apache License 2.0
- MMDetection[4] ..... Apache License 2.0
- MMDetection3d[5] ..... Apache License 2.0

D.2. Public Datasets Used

We acknowledge the use of the following public datasets, during the course of this work.

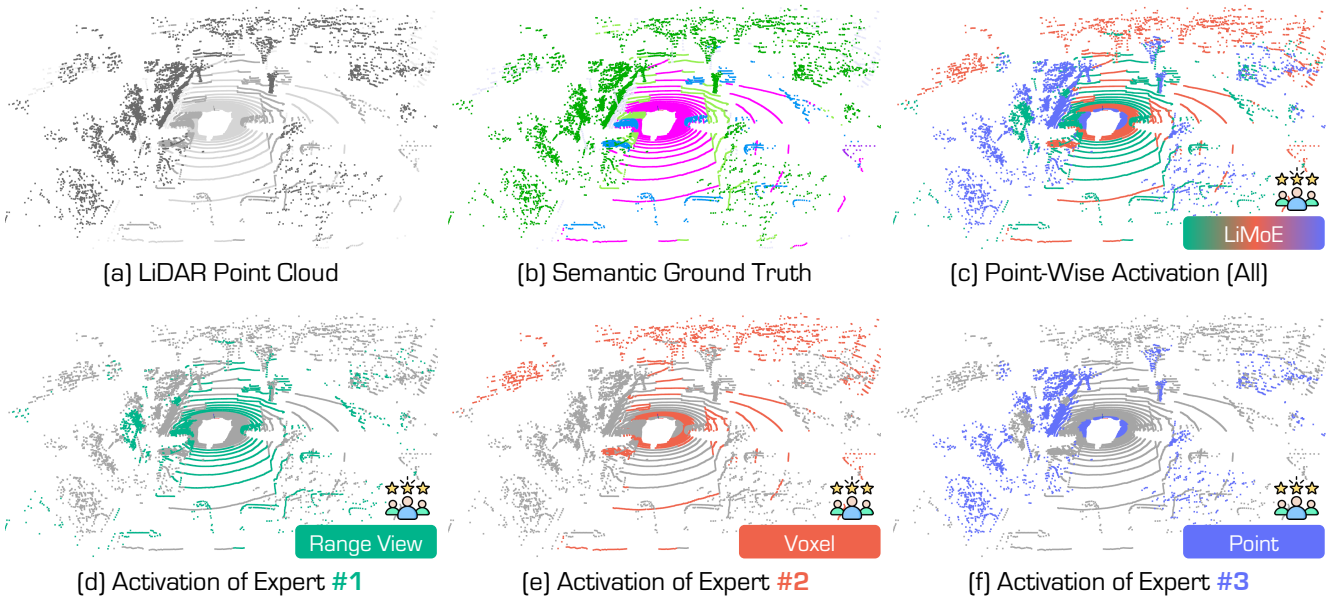
- nuScenes[6] ..... CC BY-NC-SA 4.0
- SemanticKITTI[7] ..... CC BY-NC-SA 4.0
- WaymoOpenDataset[8] ..... Waymo Dataset License
- ScribbleKITTI[9] ..... Unknown
- RELLIS-3D[10] ..... CC BY-NC-SA 3.0
- SemanticPOSS[11] ..... CC BY-NC-SA 3.0
- SemanticSTF[12] ..... CC BY-NC-SA 4.0
- SynLiDAR[13] ..... MiT License
- DAPS-3D[14] ..... MiT License
- Synth4D[15] ..... GPL-3.0 License
- Robo3D[16] ..... CC BY-NC-SA 4.0

D.3. Public Implementations Used

We acknowledge the use of the following public implementations, during the course of this work.

- nuscenes-devkit[17] ..... Apache License 2.0
- semantic-kitti-api[18] ..... MIT License
- waymo-open-dataset[19] ..... Apache License 2.0
- SLidR[20] ..... Apache License 2.0
- SuperFlow[21] ..... Apache License 2.0
- FRNet[22] ..... Apache License 2.0
- DINOv2[23] ..... Apache License 2.0
- torchsparse[24] ..... MIT License
- Conv-LoRA[25] ..... Apache License 2.0
- MoE-LLaVA[26] ..... Apache License 2.0

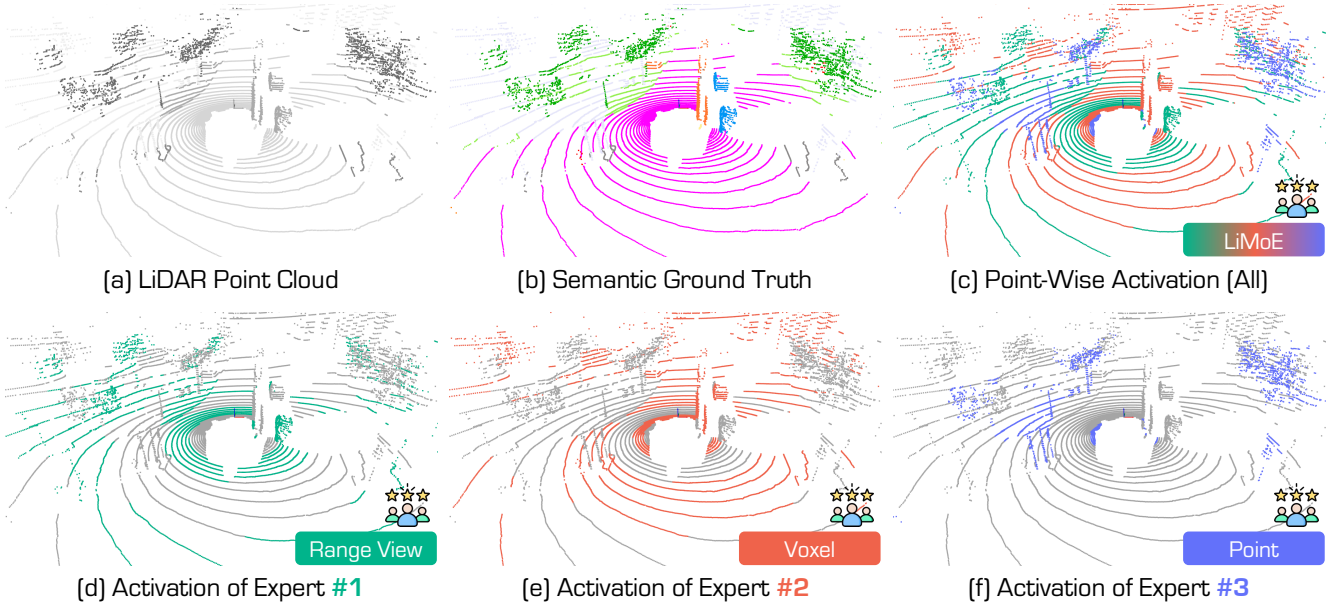
[1] https://github.com/open-mmlab/mmengine.
[2] https://github.com/open-mmlab/mmcv.
[3] https://github.com/open-mmlab/mmpretrain.
[4] https://github.com/open-mmlab/mmdetection.
[5] https://github.com/open-mmlab/mmdetection3d.
[6] https://www.nuscenes.org/nuscenes.
[7] http://semantic-kitti.org.
[8] https://waymo.com/open.
[9] https://github.com/ouenal/scribblekitti.
[10] https://github.com/unmannedlab/RELLIS-3D.
[11] http://www.poss.pku.edu.cn/semanticposs.html.
[12] https://github.com/xiaoaoran/SemanticSTF.
[13] https://github.com/xiaoaoran/SynLiDAR.
[14] https://github.com/subake/DAPS3D.
[15] https://github.com/saltoricristiano/gipso-sfouda.
[16] https://github.com/ldkong1205/Robo3D.
[17] https://github.com/nutonomy/nuscenes-devkit.
[18] https://github.com/PRBonn/semantic-kitti-api.
[19] https://github.com/waymo-research/waymo-open-dataset.
[20] https://github.com/valeoai/SLidR.
[21] https://github.com/Xiangxu-0103/SuperFlow.
[22] https://github.com/Xiangxu-0103/FRNet.
[23] https://github.com/facebookresearch/dinov2.
[24] https://github.com/mit-han-lab/torchsparse.
[25] https://github.com/autogluon/autogluon.
[26] https://github.com/PKU-YuanGroup/MoE-LLaVA.



(a) LiDAR Point Cloud (b) Semantic Ground Truth (c) Point-Wise Activation (All)

(d) Activation of Expert #1 (e) Activation of Expert #2 (f) Activation of Expert #3

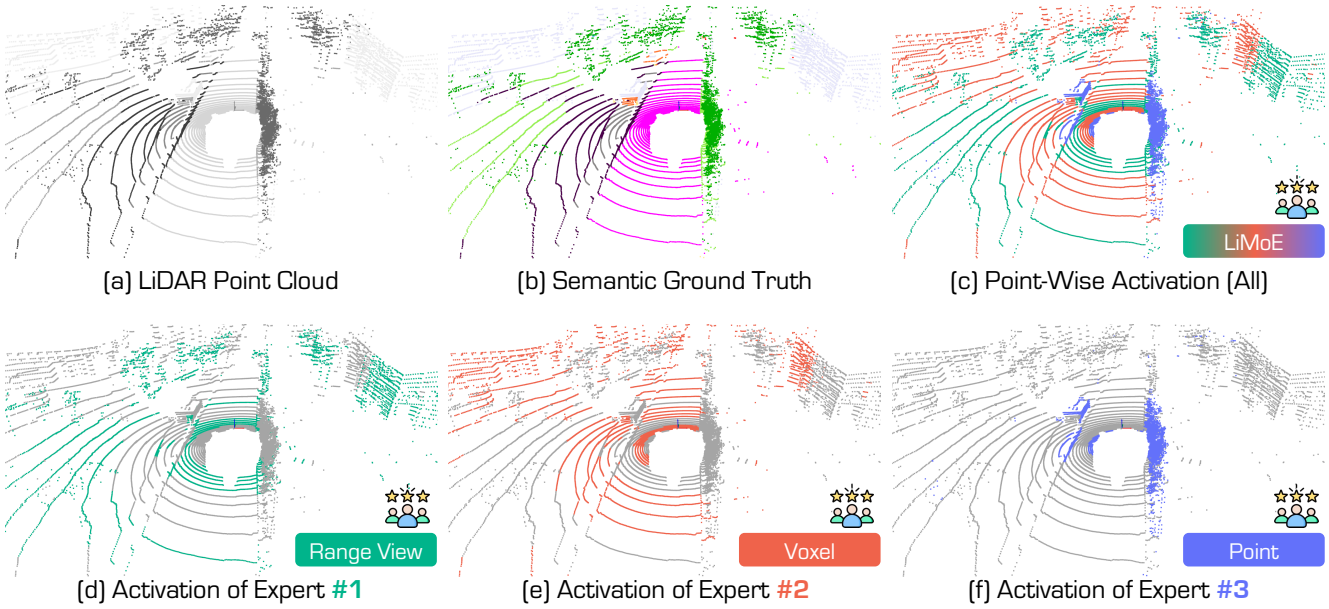
Figure C. Point-wise top-1 activation path in the SMS stage. It highlights the most activated representation for each point during the SMS stage, illustrating how different representations contribute to semantic segmentation based on spatial and object-specific characteristics. Best viewed in colors.



(a) LiDAR Point Cloud (b) Semantic Ground Truth (c) Point-Wise Activation (All)

(d) Activation of Expert #1 (e) Activation of Expert #2 (f) Activation of Expert #3

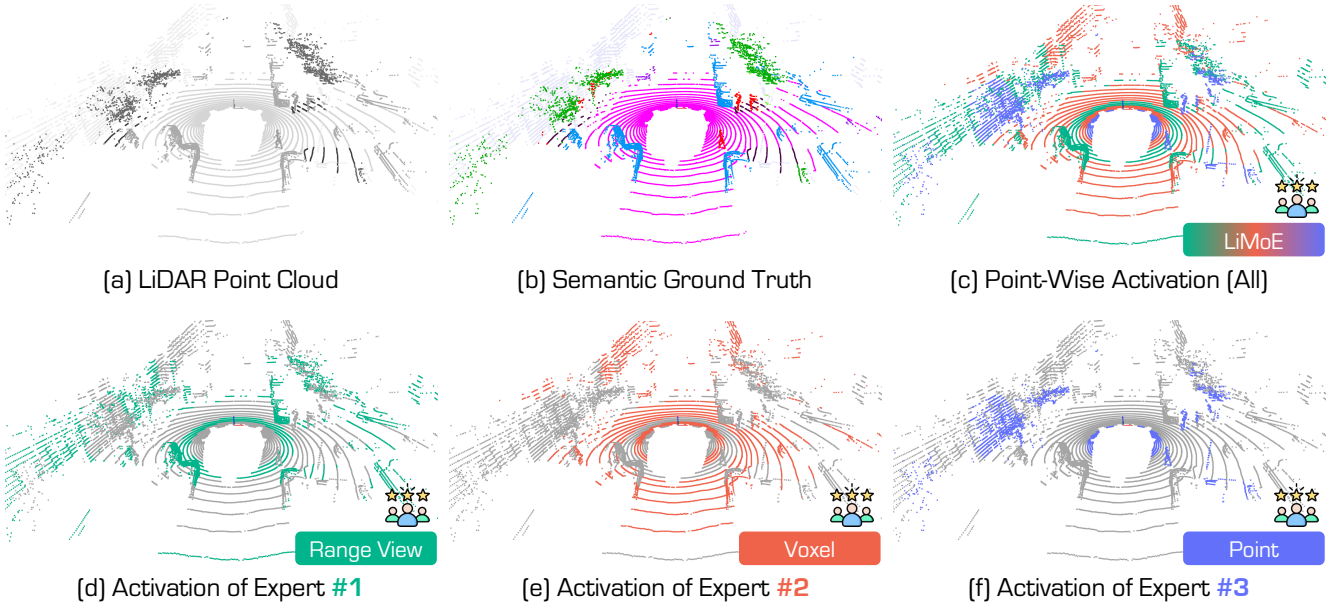
Figure D. Point-wise top-1 activation path in the SMS stage. It highlights the most activated representation for each point during the SMS stage, illustrating how different representations contribute to semantic segmentation based on spatial and object-specific characteristics. Best viewed in colors.



(a) LiDAR Point Cloud (b) Semantic Ground Truth (c) Point-Wise Activation (All)

(d) Activation of Expert #1 (e) Activation of Expert #2 (f) Activation of Expert #3

Figure E. Point-wise top-1 activation path in the SMS stage. It highlights the most activated representation for each point during the SMS stage, illustrating how different representations contribute to semantic segmentation based on spatial and object-specific characteristics. Best viewed in colors.



(a) LiDAR Point Cloud (b) Semantic Ground Truth (c) Point-Wise Activation (All)

(d) Activation of Expert #1 (e) Activation of Expert #2 (f) Activation of Expert #3

Figure F. Point-wise top-1 activation path in the SMS stage. It highlights the most activated representation for each point during the SMS stage, illustrating how different representations contribute to semantic segmentation based on spatial and object-specific characteristics. Best viewed in colors.

Table D. The **per-class IoU scores** of state-of-the-art pretraining methods pretrained and linear-probed on the *nuScenes* [5, 9] dataset. All scores are given in percentage (%). The **best** and 2nd best scores under each group are highlighted in **bold** and underline.

Method	mIoU	barrier	bicycle	bus	car	construction vehicle	motorcycle	pedestrian	traffic cone	trailer	truck	driveable surface	other flat	sidewalk	terrain	manmade	vegetation
Random	8.1	0.5	0.0	0.0	3.9	0.0	0.0	0.0	6.4	0.0	3.9	59.6	0.0	0.1	16.2	30.6	12.0
Distill: None																	
PointContrast [34]	21.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DepthContrast [40]	**22.1**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ALSO [4]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BEVContrast [27]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Distill: ResNet-50																	
PPKT [18]	35.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SLidR [26]	39.2	44.2	0.0	**30.8**	60.2	15.1	22.4	47.2	27.7	16.3	34.3	80.6	21.8	35.2	48.1	71.0	71.9
ST-SLidR [20]	40.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TriCC [23]	38.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Seal [17]	45.0	**54.7**	**5.9**	30.6	**61.7**	**18.9**	**28.8**	**48.1**	**31.0**	**22.1**	**39.5**	**83.8**	**35.4**	**46.7**	**56.9**	**74.7**	**74.7**
CSC [6]	**46.0**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
HVDistill [39]	39.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Distill: ViT-S																	
PPKT [18]	38.6	43.8	0.0	31.2	53.1	15.2	0.0	42.2	16.5	18.3	33.7	79.1	37.2	45.2	52.7	75.6	74.3
SLidR [26]	44.7	45.0	8.2	34.8	58.6	23.4	40.2	43.8	19.0	22.9	40.9	82.7	38.3	47.6	53.9	77.8	77.9
+ LiMoE	45.8	46.2	**8.5**	36.2	59.4	**23.6**	**41.7**	47.2	20.7	24.1	43.2	83.9	38.7	**48.1**	55.3	**78.0**	78.6
Seal [17]	45.2	48.9	8.4	30.7	**68.1**	17.5	37.7	57.7	17.9	20.9	40.4	83.8	36.6	44.2	54.5	76.2	**79.3**
SuperFlow [35]	46.4	49.8	6.8	45.9	63.4	18.5	31.0	60.3	28.1	25.4	47.4	86.2	38.4	47.4	56.7	74.9	77.8
+ LiMoE	**48.2**	**50.4**	7.9	**46.7**	65.1	19.2	32.1	**61.5**	**29.5**	**26.7**	**48.3**	**86.5**	**39.1**	48.0	**57.4**	75.1	78.4
Distill: ViT-B																	
PPKT [18]	40.0	29.6	0.0	30.7	55.8	6.3	22.4	56.7	18.1	24.3	42.7	82.3	33.2	45.1	53.4	71.3	75.7
SLidR [26]	45.4	46.7	7.8	46.5	58.7	23.9	34.0	47.8	17.1	23.7	41.7	83.4	39.4	47.0	54.6	76.6	77.8
+ LiMoE	46.6	48.2	8.6	47.1	61.1	**25.0**	35.3	48.6	18.4	**24.4**	43.4	84.6	**39.9**	47.4	56.9	**77.4**	78.9
Seal [17]	46.6	**49.3**	8.2	35.1	**70.8**	22.1	41.7	57.4	15.2	21.6	42.6	84.5	38.1	46.8	55.4	77.2	**79.5**
SuperFlow [35]	47.7	45.8	12.4	52.6	67.9	17.2	40.8	59.5	25.4	21.0	47.6	85.8	37.2	48.4	56.6	76.2	78.2
+ LiMoE	**49.1**	46.8	**13.1**	**53.9**	68.4	19.2	**42.2**	**59.9**	**27.5**	21.7	**48.3**	**85.9**	38.2	**49.0**	**57.1**	76.3	78.8
Distill: ViT-L																	
PPKT [18]	41.6	30.5	0.0	32.0	57.3	8.7	24.0	58.1	19.5	**24.9**	44.1	83.1	34.5	45.9	55.4	72.5	76.4
SLidR [26]	45.7	46.9	6.9	44.9	60.8	22.7	40.6	44.7	17.4	23.0	40.4	83.6	39.9	47.8	55.2	78.1	78.3
+ LiMoE	47.4	48.7	9.2	46.7	62.7	**24.2**	42.1	46.2	19.7	24.4	43.2	**85.3**	**41.6**	**49.5**	57.4	**78.7**	79.3
Seal [17]	46.8	53.1	6.9	35.0	65.0	22.0	46.1	**59.2**	16.2	23.0	41.8	84.7	35.8	46.6	55.5	78.4	**79.8**
SuperFlow [35]	48.0	52.3	12.7	46.5	64.7	21.4	44.9	56.2	26.7	19.9	43.2	84.2	38.1	47.4	56.9	76.0	79.2
+ LiMoE	**49.4**	**54.4**	**14.4**	**47.9**	**66.1**	23.9	**46.7**	57.2	**27.9**	20.8	**44.8**	85.0	39.6	48.1	**58.2**	76.5	79.6

Table E. The **per-class IoU scores** of state-of-the-art pretraining methods pretrained and fine-tuned on the *nuScenes* [5, 9] dataset with 1% annotations. All scores are given in percentage (%). The **best** and 2nd best scores under each group are highlighted in **bold** and underline.

Method	mIoU	barrier	bicycle	bus	car	construction vehicle	motorcycle	pedestrian	traffic cone	trailer	truck	driveable surface	other flat	sidewalk	terrain	manmade	vegetation
Random	30.3	0.0	0.0	8.1	65.0	0.1	6.6	21.0	9.0	9.3	25.8	89.5	14.8	41.7	48.7	72.4	73.3
Distill: None																	
PointContrast [34]	32.5	0.0	1.0	5.6	67.4	0.0	3.3	31.6	5.6	12.1	30.8	**91.7**	21.9	**48.4**	50.8	75.0	74.6
DepthContrast [40]	31.7	0.0	0.6	6.5	64.7	0.2	**5.1**	29.0	**9.5**	12.1	29.9	90.3	17.8	44.4	49.5	73.5	74.0
ALSO [4]	37.7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BEVContrast [27]	**37.9**	0.0	**1.3**	**32.6**	**74.3**	**1.1**	0.9	**41.3**	8.1	**24.1**	**40.9**	89.8	**36.2**	44.0	**52.1**	**79.9**	**79.7**
Distill: ResNet-50																	
PPKT [18]	37.8	0.0	2.2	20.7	75.4	1.2	13.2	45.6	8.5	17.5	38.4	92.5	19.2	52.3	56.8	80.1	80.9
SLidR [26]	38.8	0.0	1.8	15.4	73.1	1.9	19.9	47.2	17.1	14.5	34.5	92.0	27.1	53.6	61.0	79.8	82.3
ST-SLidR [20]	40.8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TriCC [23]	41.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Seal [17]	45.8	0.0	**9.4**	32.6	**77.5**	**10.4**	28.0	53.0	25.0	**30.9**	**49.7**	**94.0**	**33.7**	**60.1**	59.6	**83.9**	83.4
CSC [6]	**47.0**	0.0	0.0	**58.7**	74.0	0.1	**40.9**	**58.9**	**31.8**	23.7	45.1	92.5	33.0	56.4	**62.4**	81.6	**84.2**
HVDistill [39]	42.7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Distill: ViT-S																	
PPKT [18]	40.6	0.0	0.0	25.2	73.5	9.1	6.9	51.4	8.6	11.3	31.1	93.2	**41.7**	58.3	64.0	82.0	82.6
SLidR [26]	41.2	0.0	0.0	26.6	72.0	12.4	15.8	51.4	22.9	11.7	35.3	92.9	36.3	58.7	63.6	81.2	82.3
+ LiMoE	46.8	20.6	4.2	**29.7**	74.7	16.9	24.6	55.7	28.3	19.5	41.5	93.8	41.0	**62.4**	**67.3**	82.6	**85.2**
Seal [17]	44.3	20.0	0.0	19.4	74.7	10.6	**45.7**	60.3	**29.2**	17.4	38.1	93.2	26.0	58.8	64.5	81.9	81.9
SuperFlow [35]	47.8	38.2	1.8	25.8	79.0	15.3	43.6	60.3	0.0	28.4	55.4	93.7	28.8	59.1	59.9	83.5	83.1
+ LiMoE	**49.6**	**39.9**	**4.6**	27.3	**80.2**	**17.1**	45.4	**61.2**	6.2	**29.5**	**58.4**	**94.0**	34.2	62.3	64.6	**84.1**	84.5
Distill: ViT-B																	
PPKT [18]	40.9	0.0	0.0	24.5	73.5	12.2	7.0	51.0	13.5	15.4	36.3	93.1	40.4	59.2	63.5	81.7	82.2
SLidR [26]	41.6	0.0	0.0	26.7	73.4	10.3	16.9	51.3	23.3	12.7	38.1	93.0	37.7	58.8	63.4	81.6	82.7
+ LiMoE	46.9	22.7	2.6	28.3	75.4	**13.5**	27.8	55.0	**28.5**	22.2	40.6	93.7	**42.3**	61.9	**66.8**	83.1	**85.4**
Seal [17]	46.0	**43.0**	0.0	26.7	81.3	9.9	41.3	56.2	0.0	21.7	51.6	93.6	**42.3**	62.8	64.7	82.6	82.7
SuperFlow [35]	48.1	39.1	0.9	30.0	80.7	10.3	47.1	59.5	5.1	27.6	55.4	93.7	29.1	61.1	63.5	82.7	83.6
+ LiMoE	**50.2**	41.5	**3.8**	**32.2**	**81.7**	12.9	**49.3**	**61.1**	7.3	**29.3**	**57.8**	**94.2**	35.1	**62.9**	65.4	**84.0**	84.8
Distill: ViT-L																	
PPKT [18]	42.1	0.0	0.0	24.4	78.8	15.1	9.2	54.2	14.3	12.9	39.1	92.9	37.8	59.8	64.9	82.3	83.6
SLidR [26]	42.8	0.0	0.0	23.9	78.8	15.2	20.9	55.0	28.0	17.4	41.4	92.2	41.2	58.0	64.0	81.8	82.7
+ LiMoE	46.9	21.6	1.6	25.2	80.1	17.3	28.0	56.4	**28.3**	18.6	43.1	92.7	**41.7**	60.9	**65.5**	83.8	**85.6**
Seal [17]	46.3	41.8	0.0	23.8	81.4	17.7	46.3	58.6	0.0	23.4	54.7	93.8	41.4	62.5	65.0	83.9	83.8
SuperFlow [35]	50.0	44.5	0.9	22.4	80.8	17.1	50.2	60.9	21.0	25.1	55.1	93.9	35.8	61.5	62.6	83.7	83.7
+ LiMoE	**51.4**	**45.3**	**4.1**	**25.3**	**82.2**	**18.4**	**52.5**	**61.8**	22.3	**26.4**	**56.2**	**94.3**	37.6	**63.3**	63.9	**84.4**	85.0

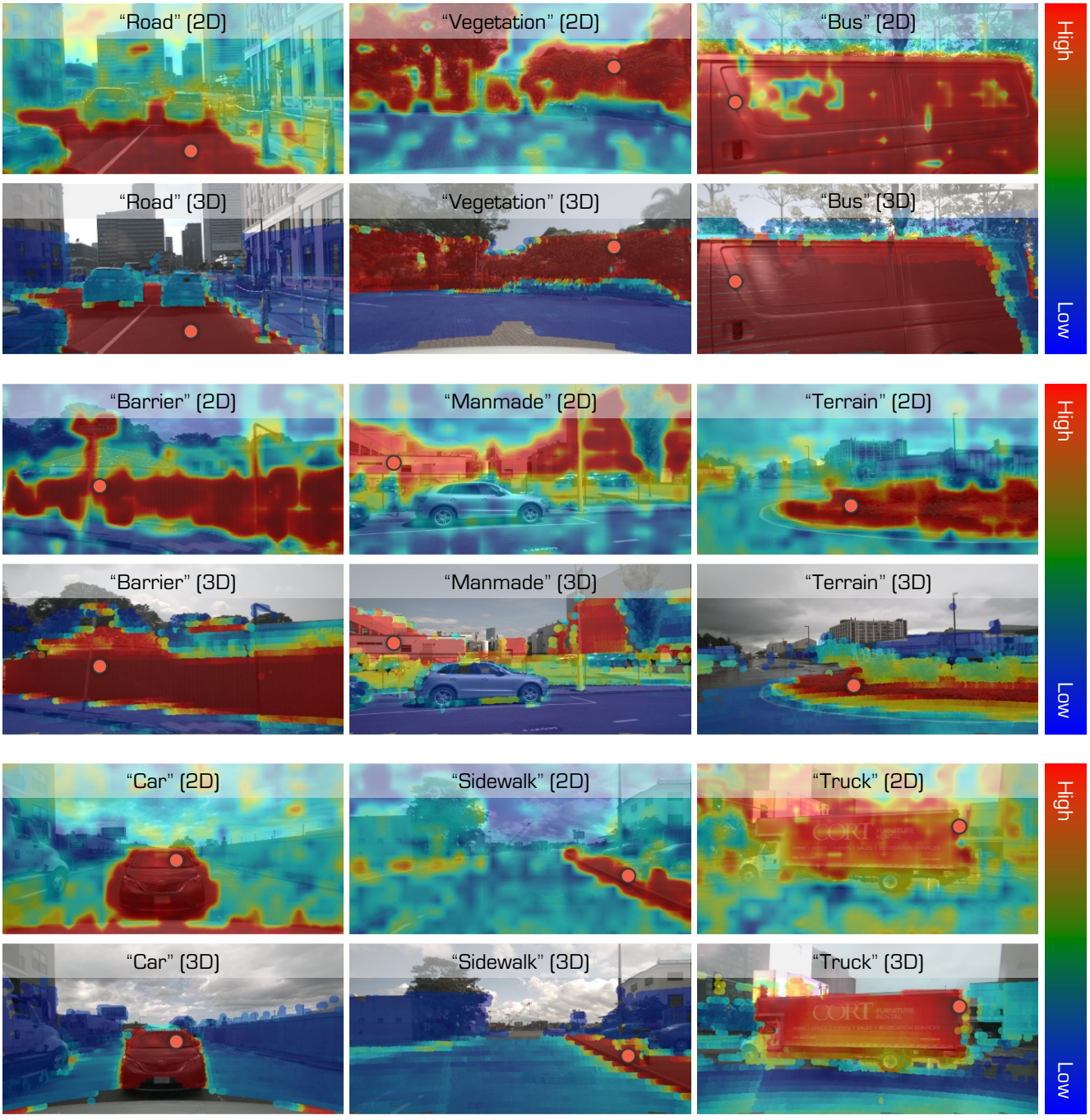


Figure G. **Cosine similarity** between the learned features of a query point (denoted as the red dot) and: (1) the features of the image of the same scene (the 1st, 3rd, and 5th rows); and (2) the features of the LiDAR points of the same scene that are projected onto the image (the 2nd, 4th, and 6th rows). Best viewed in colors.

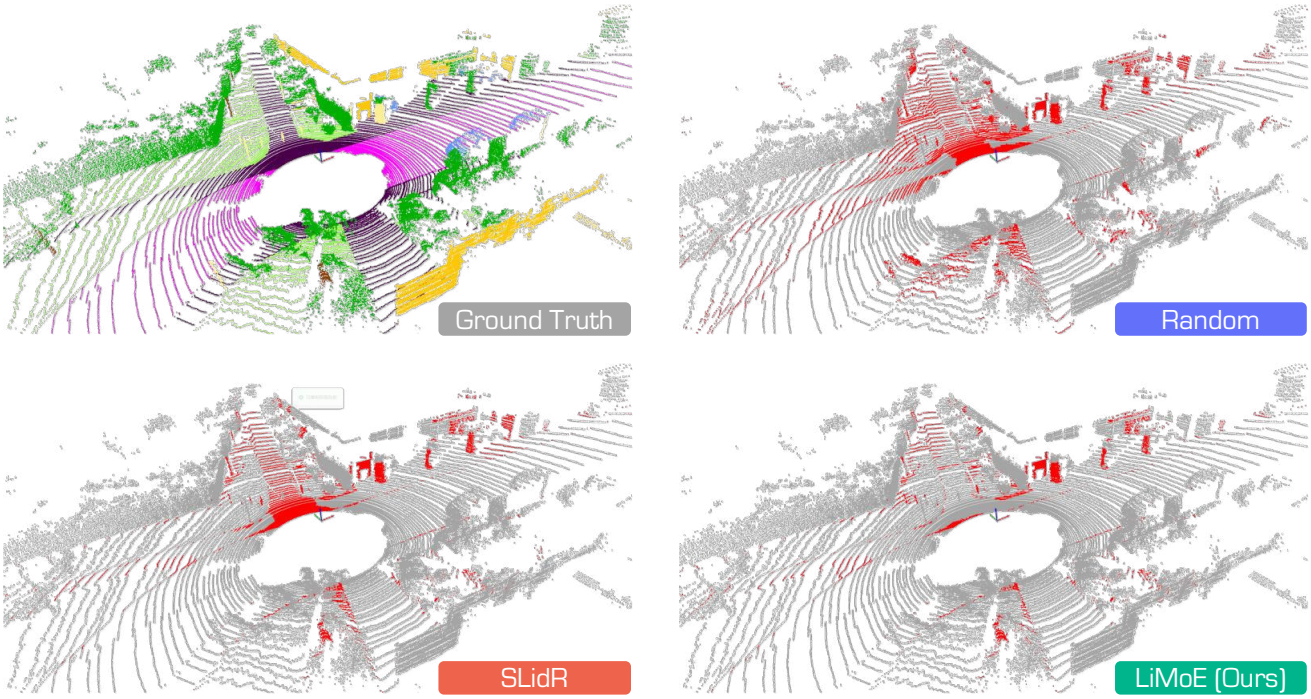


Figure H. **Qualitative assessments** of state-of-the-art pretraining methods, pretrained on *nuScenes* [5] and fine-tuned on *SemanticKITTI* [1] with 1% annotations. The error maps depict correct and incorrect predictions in gray and red, respectively. Best viewed in colors.

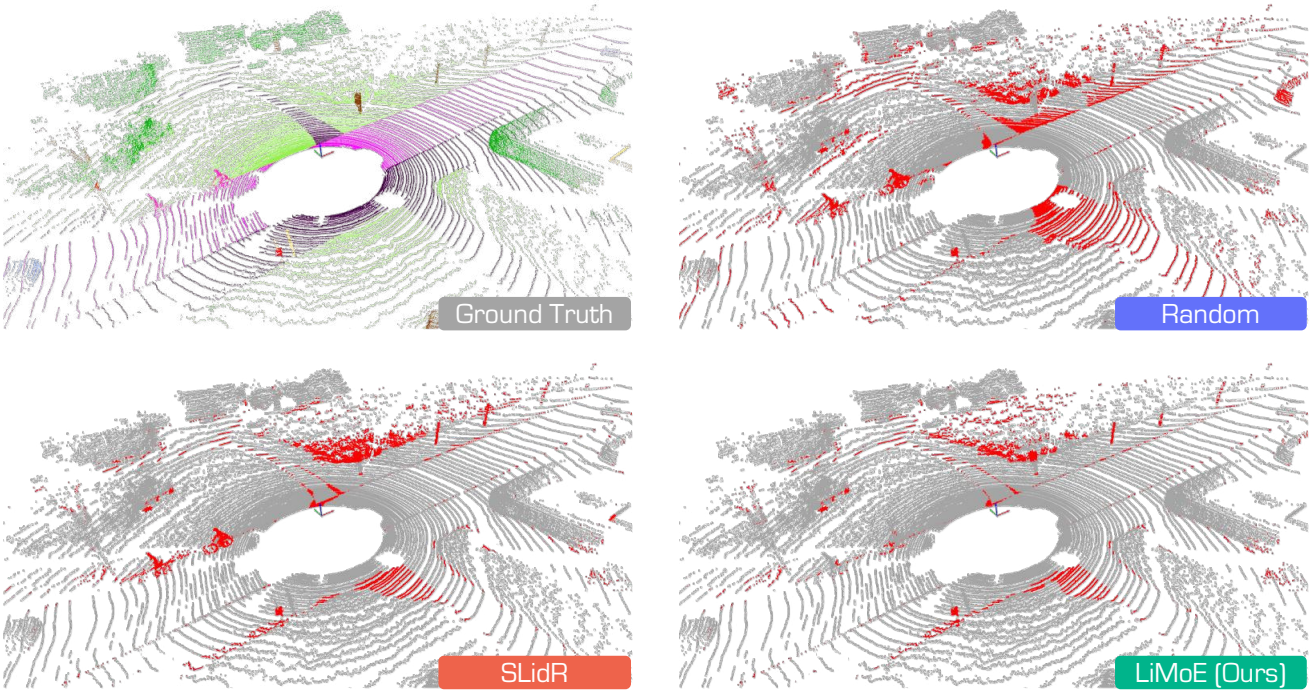


Figure I. **Qualitative assessments** of state-of-the-art pretraining methods, pretrained on *nuScenes* [5] and fine-tuned on *SemanticKITTI* [1] with 1% annotations. The error maps depict correct and incorrect predictions in gray and red, respectively. Best viewed in colors.

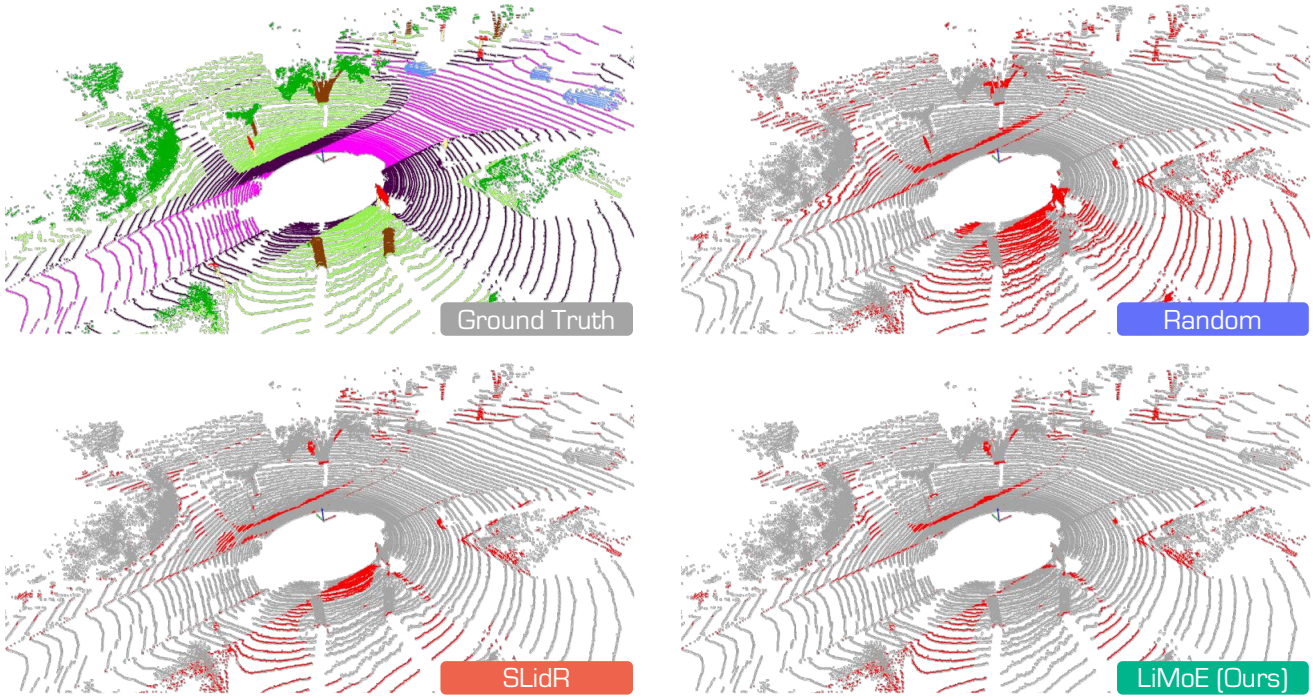


Figure J. **Qualitative assessments** of state-of-the-art pretraining methods, pretrained on *nuScenes* [5] and fine-tuned on *SemanticKITTI* [1] with 1% annotations. The error maps depict correct and incorrect predictions in gray and red, respectively. Best viewed in colors.

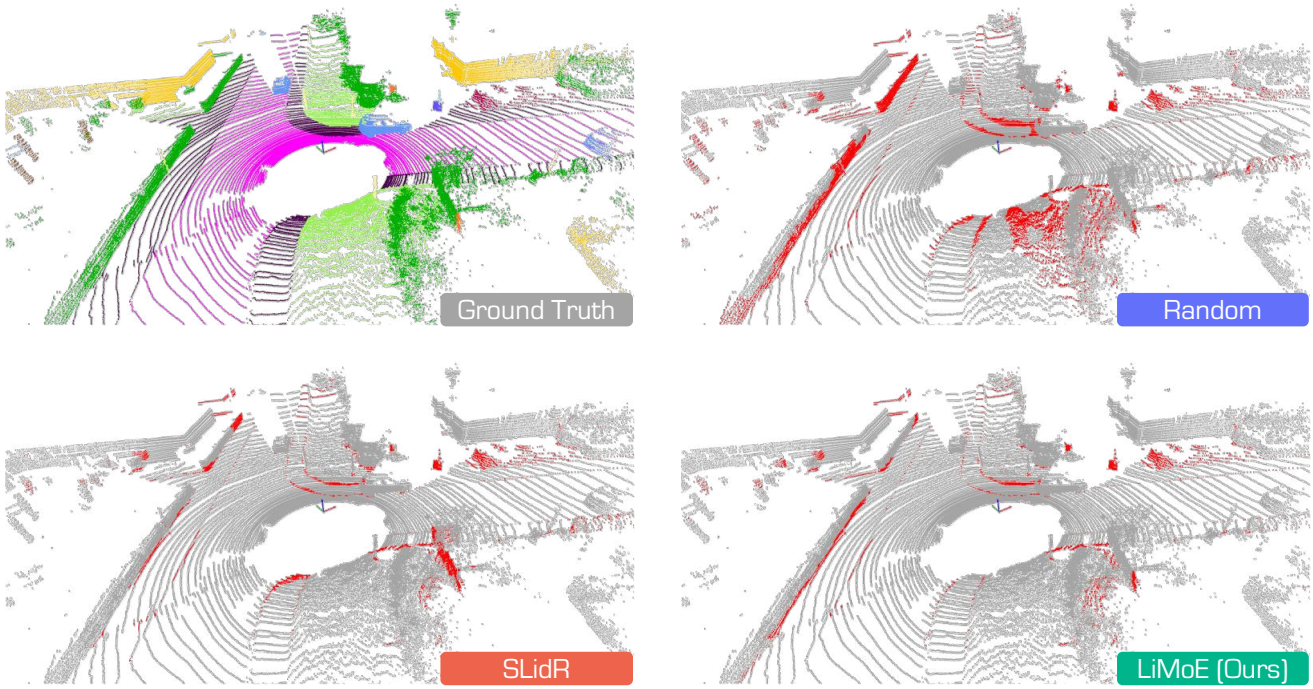


Figure K. **Qualitative assessments** of state-of-the-art pretraining methods, pretrained on *nuScenes* [5] and fine-tuned on *SemanticKITTI* [1] with 1% annotations. The error maps depict correct and incorrect predictions in gray and red, respectively. Best viewed in colors.

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019. 1, 4, 6, 13, 14

[2] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018. 3

[3] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11682–11692, 2020. 2

[4] Alexandre Boulch, Corentin Sautier, Björn Michele, Gilles Puy, and Renaud Marlet. Also: Automotive lidar self-supervision by occupancy estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13455–13465, 2023. 10, 11

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 1, 4, 10, 11, 13, 14

[6] Haoming Chen, Zhizhong Zhang, Yanyun Qu, Ruixin Zhang, Xin Tan, and Yuan Xie. Building a strong pre-training baseline for universal 3d large-scale perception. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19925–19935, 2024. 4, 10, 11

[7] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 2, 5

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2

[9] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *IEEE Robotics and Automation Letters*, 7:3795–3802, 2022. 1, 4, 6, 10, 11

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 1

[11] Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripalli. Rellis-3d dataset: Data, benchmarks and analysis. In *IEEE International Conference on Robotics and Automation*, pages 1110–1116, 2021. 1

[12] Alexey A Klokov, Di Un Pak, Aleksandr Khorin, Dmitry A Yudin, Leon Kochiev, Vladimir D Luchinskiy, and Vitaly D Bezuglyj. Daps3d: Domain adaptive projective segmentation of 3d lidar point clouds. *IEEE Access*, 11:79341–79356, 2023. 2

[13] Lingdong Kong, Youquan Liu, Xin Li, Runnan Chen, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Robo3d: Towards robust and reliable 3d perception against corruptions. In *IEEE/CVF International Conference on Computer Vision*, pages 19994–20006, 2023. 2, 3

[14] Lingdong Kong, Jiawei Ren, Liang Pan, and Ziwei Liu. Lasermix for semi-supervised lidar semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21705–21715, 2023. 6

[15] Lingdong Kong, Xiang Xu, Jiawei Ren, Wenwei Zhang, Liang Pan, Kai Chen, Wei Tsang Ooi, and Ziwei Liu. Multimodal data-efficient 3d scene understanding for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 6

[16] Hanxue Liang, Chenhan Jiang, Dapeng Feng, Xin Chen, Hang Xu, Xiaodan Liang, Wei Zhang, Zhenguo Li, and Luc Van Gool. Exploring geometry-aware contrast and clustering harmonization for self-supervised 3d object detection. In *IEEE/CVF International Conference on Computer Vision*, pages 3293–3302, 2021. 4

[17] Youquan Liu, Lingdong Kong, Jun Cen, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. Segment any point cloud sequences by distilling vision foundation models. In *Advances in Neural Information Processing Systems*, pages 37193–37229, 2023. 10, 11

[18] Yueh-Cheng Liu, Yu-Kai Huang, Hung-Yueh Chiang, Hung-Ting Su, Zhe-Yu Liu, Chin-Tang Chen, Ching-Yu Tseng, and Winston H Hsu. Learning from 2d: Contrastive pixel-to-point knowledge transfer for 3d pretraining. *arXiv preprint arXiv:2104.0468*, 2021. 10, 11

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 2, 3

[20] Anas Mahmoud, Jordan SK Hu, Tianshu Kuai, Ali Harakeh, Liam Paull, and Steven L Waslander. Self-supervised image-to-point distillation via semantically tolerant contrastive loss. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7102–7110, 2023. 10, 11

[21] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2

[22] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. Semanticposs: A point cloud dataset with large quantity of dynamic instances. In *IEEE Intelligent Vehicles Symposium*, pages 687–693, 2020. 2

[23] Bo Pang, Hongchi Xia, and Cewu Lu. Unsupervised 3d point cloud representation learning by triangle constrained contrast for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5229–5239, 2023. 4, 10, 11

[24] Ryan Razani, Ran Cheng, Ehsan Taghavi, and Liu Bingbing. Lite-hdseg: Lidar semantic segmentation using lite harmonic dense convolutions. In *IEEE International Conference on Robotics and Automation*, pages 9550–9556, 2021. 3

[25] Cristiano Saltori, Evgeny Krivosheev, Stéphane Lathuiliére, Nicu Sebe, Fabio Galasso, Giuseppe Fiameni, Elisa Ricci, and Fabio Poiesi. Gipso: Geometrically informed propagation for online adaptation in 3d lidar segmentation. In *European Conference on Computer Vision*, pages 567–585, 2022. 2

[26] Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-lidar self-supervised distillation for autonomous driving data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9891–9901, 2022. 1, 2, 4, 5, 10, 11

[27] Corentin Sautier, Gilles Puy, Alexandre Boulch, Renaud Marlet, and Vincent Lepetit. Bevcontrast: Self-supervision in bev space for automotive lidar point clouds. In *International Conference on 3D Vision*, pages 559–568, 2024. 10, 11

[28] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. *arXiv preprint arXiv:1708.07120*, 2017. 2, 3

[29] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 1

[30] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *European Conference on Computer Vision*, pages 685–702, 2020. 2, 5

[31] Ozan Unal, Dengxin Dai, and Luc Van Gool. Scribble-supervised lidar semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2697–2707, 2022. 1

[32] Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu. Transfer learning from synthetic to real lidar point cloud for semantic segmentation. In *AAAI Conference on Artificial Intelligence*, pages 2795–2803, 2022. 2

[33] Aoran Xiao, Jiaxing Huang, Weihao Xuan, Ruijie Ren, Kangcheng Liu, Dayan Guan, Abdulmotaleb El Saddik, Shijian Lu, and Eric P Xing. 3d semantic segmentation in the wild: Learning generalized models for adverse-condition point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9382–9392, 2023. 2

[34] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European Conference on Computer Vision*, pages 574–591, 2020. 4, 10, 11

[35] Xiang Xu, Lingdong Kong, Hui Shuai, Wenwei Zhang, Liang Pan, Kai Chen, Ziwei Liu, and Qingshan Liu. 4d contrastive superflows are dense 3d representation learners. In *European Conference on Computer Vision*, pages 58–80, 2024. 2, 4, 10, 11

[36] Xiang Xu, Lingdong Kong, Hui Shuai, and Qingshan Liu. Frnet: Frustum-range networks for scalable lidar segmentation. *IEEE Transactions on Image Processing*, 2025. 2, 5

[37] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 4

[38] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11784–11793, 2021. 4, 5

[39] Sha Zhang, Jiajun Deng, Lei Bai, Houqiang Li, Wanli Ouyang, and Yanyong Zhang. Hvdistill: Transferring knowledge from images to point clouds via unsupervised hybrid-view distillation. *International Journal of Computer Vision*, 132:2585—2599, 2024. 10, 11

[40] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *IEEE/CVF International Conference on Computer Vision*, pages 10252–10263, 2021. 10, 11

[41] Zixiang Zhou, Yang Zhang, and Hassan Foroosh. Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13194–13203, 2021. 6